1. program -> declaration-list **EOF**
2. declaration-list -> declaration declaration-list | eps
3. declaration -> type-specifier **ID** declaration-2
   1. declaration-2 -> **;** | **[** **NUM** **]** **;** | **(** params **)** compound-stmt
4. type-specifier -> **int** | **void**
5. params -> **int** **ID** param-2 params-list-2 | **void** params-2
   1. params-2 -> **ID** param-2 params-list-2 | eps
6. params-list -> param params-list-2
   1. params-list-2 ->**,** params-list | eps
7. param -> type-specifier **ID** param-2
   1. param-2 -> **[** **]** | eps
8. compound-stmt -> { declaration-list statement-list }
9. statement-list -> statement statement-list | eps
10. statement -> expression-stmt | compound-stmt | selection-stmt | iteration-stmt | return-stmt | switch-stmt
11. expression-stmt -> expression ; | continue ; | break ; | ;
12. selection-stmt -> if ( expression ) statement else statement
13. iteration-stmt -> while ( expression ) statement
14. return-stmt -> return return-stmt-2
    1. return-stmt-2 -> ; | expression ;
15. switch-stmt -> switch ( expression ) { case-stmts default-stmt }
16. case-stmts -> case-stmt case-stmts | eps
17. case-stmt -> case NUM : statement-list
18. default-stmt -> default : statement-list | eps
19. var -> ID var-2
    1. var-2 -> [ expression ] | eps
20. simple-expression -> additive-expression simple-expression-2
    1. simple-expression-2 -> relop additive-expression | eps
21. relop -> < | ==
22. additive-expression -> term additive-expression-2
    1. additive-expression-2 -> addop additive-expression | eps
23. addop -> + | -
24. term -> signed-factor term-2
    1. term-2 -> eps | \* term
25. signed-factor -> factor | + factor | - factor
26. factor -> ( expression ) | NUM | ID factor-2
    1. factor-2 -> [ expression ] | ( args ) | eps
27. args -> arg-list | eps
28. arg-list -> expression arg-list-2
    1. arg-list-2 -> , arg-list | eps
29. expression -> + factor term-2 additive-expression-2 simple-expression-2 |   
     - factor term-2 additive-expression-2 simple-expression-2 |   
     ( expression ) term-2 additive-expression-2 simple-expression-2 |  
     NUM term-2 additive-expression-2 simple-expression-2 |   
     ID expression-2
    1. expression-2 -> [ expression ] expression-3 | expression-3 | ( args ) term-2 additive-expression-2 simple-expression-2
    2. expression-3 -> \* term additive-expression-2 simple-expression-2 | < additive-expression | addop additive-expression simple-expression-2 | = expression-4 | eps
    3. expression-4 -> expression | = additive-expression